

Amendment to the Claims

1. (currently amended) A computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state, said method comprising:

receiving a child process having inherited an instrumented parent process' context including the parent's program text that may have been modified by instrumentation;

modifying selected text segment portions from said child process;

unmapping instrumented code space such that said instrumented code space is inaccessible to said child process;

provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to uninstrumented code space; and

executing said process and, provided said child process generates a fault by seeking to access an address in instrumented code space, providing a corresponding address in said uninstrumented code space.

2. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said selected text segment portions are selected from the group consisting of: breakpoints, branches, switch tables, procedure lookup tables (PLTs) for said instrumented code space.

3. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said instrumented code space is comprised of shared memory.

4. (currently amended) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 further comprising: unwinding a call stack of said child process and recording return addresses of said child process.

5. (currently amended) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 4 further comprising: comparing said return addresses of said child process to said address in said instrumented code space which generated said fault upon execution of said child process.

6. (currently amended) A computer-readable medium embodying instructions that cause a computer to perform a method for reverting a process in an in-line instrumented state to an uninstrumented state, the method comprising:  
receiving a child process having inherited an instrumented parent process' context including the parent's program text that may have been modified by instrumentation;

modifying selected text segment portions from said child process;

unmapping instrumented code space such that said instrumented code space is inaccessible to said child process;

provided an instruction pointer resides in said instrumented code space, updating said instruction pointer to uninstrumented code space; and

executing said child process and, provided said process generates a fault by seeking to access an address in instrumented code space, providing a

corresponding address in said uninstrumented code space.

7. (original) The computer-readable medium of claim 6 wherein said selected text segment portions are selected from the group consisting of: breakpoints, branches, switch tables, procedure lookup tables (PLTs) for said instrumented code space.

8. (original) The computer-readable medium of claim 6 wherein said instrumented code space is comprised of shared memory.

9. (currently amended) The computer-readable medium of claim 6 wherein said instructions further cause said computer to: unwind a call stack of said child process and record return addresses of said child process.

10. (currently amended) The computer-readable medium of claim 9 wherein said instructions further cause said computer to: compare said return addresses of said child process to said address in said instrumented code space which generated said fault upon execution of said child process.

11. (currently amended) An apparatus for reverting a process in an in-line instrumented state to an uninstrumented state, the apparatus comprising:

means for receiving a child process having inherited an instrumented parent process' context including the parent's program text that may have been modified by instrumentation;

means for modifying selected text segment portions from said child process;

means for unmapping instrumented code space such that said instrumented code space is inaccessible to said child process;

means for updating an instruction pointer to uninstrumented code space provided said instruction pointer resides in said instrumented code space, and

means for executing said child process and, provided said process generates a fault by seeking to access an address in instrumented code space, providing a corresponding address in said uninstrumented code space.

12. (original) The apparatus of claim 11 wherein said selected text segment portions are selected from the group consisting of: breakpoints, branches, switch tables, procedure lookup tables (PLTs) for said instrumented code space.

13. (original) The apparatus of claim 11 wherein said instrumented code space is comprised of shared memory.

14. (currently amended) The apparatus of claim 11 further comprising:  
means for unwinding a call stack of said process and recording return addresses of said child process.

15. (currently amended) The apparatus of claim 14 further comprising:  
means for comparing said return addresses of said child process to said address in said instrumented code space which generated said fault upon execution of said child process.